On K-Means Cluster Preservation using Quantization Schemes

Deepak S. Turaga[‡] Michail Vlachos[†] Olivier Verscheure[‡]

[‡]IBM T.J. Watson Research Center, Hawthorne, NY, USA
[†]IBM Zürich Research Laboratory, Switzerland

Abstract

This work examines under what conditions compression methodologies can retain the outcome of clustering operations. We focus on the popular k-Means clustering algorithm and we demonstrate how a properly constructed compression scheme based on post-clustering quantization is capable of maintaining the global cluster structure. Our analytical derivation indicate that a 1-bit moment preserving quantizer per cluster is sufficient to retain the original data clusters. Merits of the proposed compression technique include: a) reduced storage requirements with clustering guarantees, b) data privacy on the original values, and c) shape preservation for data visualization purposes. We evaluate the quantization scheme on various high-dimensional datasets, including 1-dimensional and 2-dimensional time-series (shape datasets) and demonstrate the cluster preservation property. We also compare with previously proposed simplification techniques in the time-series area and show significant improvements both on the clustering and shape preservation of the compressed datasets.

1 Introduction

The exponential increase in data sizes is currently driving mining techniques into combining approximation techniques with popular knowledge extraction algorithms, in order to allow even more tractable execution times. In this study we explore how clustering algorithms could be combined with compression techniques, with the concurrent goal of providing *quality guarantees* on the clustering outcome of the approximated data. In particular, we examine *under what circumstances* the outcome of the K-Means clustering results can be preserved by compression or data simplification methods. To this end, we present a bit-quantization technique that satisfies the problem desiderata. Therefore, clustering on the simplified dataset will lead to similar results as on the uncompressed dataset. To the best of our knowledge, this is the first work that provides such guarantees for the K-Means algorithm.

Our choice for selecting K-Means as our focus of study is due to its widespread use and popularity among the data-mining and AI community. Even though many other clustering techniques with superior clustering properties have appeared (such as spectral methods [2]), K-Means is still a prevalent approach due to its many desirable properties; simplicity of implementation, amenity to parallelization and speed of execution. For applications where speed is of essence or even for doing an initial pre-clustering for data analysis, K-Means is still very much the algorithm of choice. Variations of the K-Means process are widely used as sub-processes in many analytic components.

Our approach may be intuitively described as follows. We begin by examining the objective function optimized for K-Means clustering, and then determine circumstances under which quantization or data simplification does not affect it. Since the objective function is defined in terms of the intra-cluster variance, we show that by designing a quantizer which preserves the first two moments of a time-series cluster, the objective function is preserved. We then show that this also means that the clustering outcome will be preserved. Finally, we show that using Moment Preserving Quantization (MPQ) a 1-bit quantizer per sample and class is sufficient to retain perfectly the clustering structure.

The results that we present here apply for the optimal partition of the K-Means clustering. In practice though, because of the gradient-descent based algorithm for K-Means execution (Lloyd's algorithm), and/or potentially malformed clusters on the original data, final results may deviate slightly. Our empirical results, on multiple data sets, show that clustering results are well preserved by this quantization. Importantly, we identify the cases and conditions which lead to potential discrepancies in final results.

Applications of the preservation quantization based compression scheme with K-Means cluster preservation, can find applications in the following areas:

a) **Reduced Storage**. Storing the quantized dataset requires much less space than storing the original set. This also translates to reduced transmission cost if the data set needs to be distributed. Importantly, the level of compression is tunable based on the number of bits allocated to the scalar quantizers. Other approaches that investigate scaling-up the

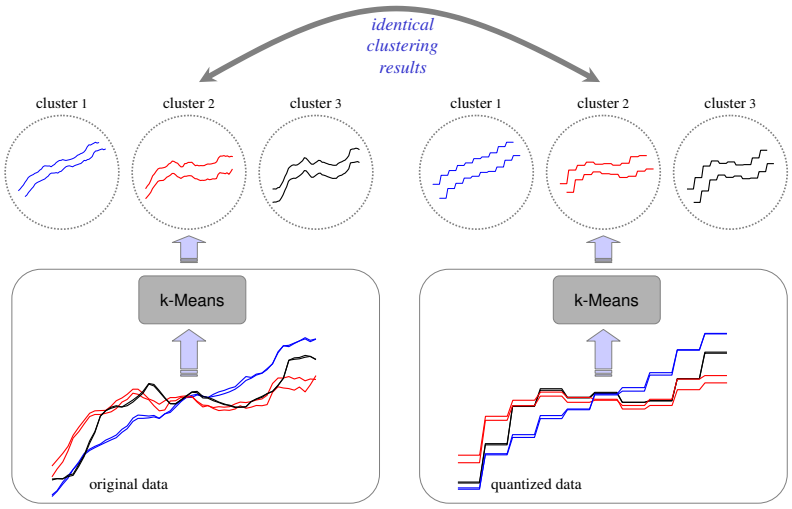

Figure 1. Objective: Design of quantization scheme for K-Means cluster preservation.

K-Means algorithm include [8, 4]. They examine the problem either from a dimensionality reduction or sampling perspective, while this work views the problem from a quantizer design angle. Additionally, these techniques make no assertions with regards to the cluster preservation.

b) **Quantization with Shape Preservation**. When the high-dimensional objects in a dataset represent time-series, (i.e., the T samples of the 1-D time series are collected into a T dimensional vector), the proposed quantization retains very closely the shape of the original sequences. This makes the compressed data amenable for a variety of mining and visualization purposes, besides the intended clustering application. In this area, our work has overlap with various time-series simplification techniques. Bagnall et al. [3] propose a *binary clipping* method for time-series data, where the data are converted into 0 and 1 if they lie above or below the mean value baseline. This representation has interesting theoretical underpinnings and has been applied for speeding up the execution of the K-Means algorithm. We compare against this work in the experimental section. Megalooikonomou et al. [9] present a piecewise vector quantized approximation for time-series data, which preserves with high accuracy the shape of the original sequences. Finally, approaches such as wavelet or Fourier approximations have been used extensively for time-series simplification, but none of these approaches are inherently designed for providing guarantees on the clustering outcome, which is one of the significant contributions of this work.

c) **Privacy Preserving Clustering**. By disseminating the quantized dataset, an added benefit is that the original values are not distributed - only sufficient approximations are revealed. Therefore our approach can also be utilized for privacy enabled mining. Vaidya et al. [12] and Jagannathan et al. [6] present privacy preserving variations for K-Means, considering the scenario when the data are segregated either vertically or horizontally. In our case, the data are not separated, but are distributed as a whole. Similar in spirit to our work are also the following efforts: Parmeswaran and Blough [11] present clustering preservation techniques through Nearest Neighbor data substitution and Oliveira and Zaane [10] present rotation based transformations (RBT) that retain the clustering outcome, by changing the object values but maintaining the pairwise object distances and hence the clustering results.

The remaining of the paper is organized as follows; we review the K-Means objective in Section 2 and describe our problem of interest in Section 3. In Section 4 we introduce a 1-bit Moment Preserving Quantization scheme (MPQ) and discuss its properties. We detail our algorithm in Section 5 and present results on real data sets in Section 6. We conclude in Section 7 with directions for future research. In the course of the paper we will describe the basic notions of our technique utilizing time-series as data objects. This is to capture in a more visual way the fundamental constructs, and also to illustrate more effectively the shape preservation property of the proposed scheme. However, the following discussion applies to any high-dimensional object.

2 Background: K-Means Clustering

Consider a set $\mathcal{S}$ consisting of N sample vectors $\mathbf{x}_j$ ($1 \leq j \leq N$), each containing T dimensions x_{ji} ($1 \leq i \leq T$). K-Means clustering involves grouping the N sample vectors into K non-overlapping clusters, i.e., subsets $\mathcal{S}_k$ ($1 \leq k \leq K$ with $\cup_k \mathcal{S}_k = \mathcal{S}$), such that the sum of intra-class variance is minimized. We may define the sum of intra-class variances as:

$$V = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{S}_k} (\mathbf{x}_j - \mu_k)^T (\mathbf{x}_j - \mu_k) \qquad (1)$$

where $\mu_k = \frac{1}{|\mathcal{S}_k|} \sum_{\mathbf{x}_j \in \mathcal{S}_k} \mathbf{x}_j$ is the centroid of each cluster. We can define the number of vectors in cluster k as $N_k = |\mathcal{S}_k|$. V is the objective function that the K-Means algorithm attempts to minimize by selecting subsets $\mathcal{S}_k$.

The objective can be expanded as follows:

$$V = \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{S}_k} \sum_{i=1}^{T} (x_{ji} - \mu_{ki})^2 \tag{2}$$

in terms of the individual dimensions x_{ji} of each object $\mathbf{x}_j$. Furthermore by swapping summations we get:

$$V = \sum_{k=1}^{K} \sum_{i=1}^{T} \sum_{\mathbf{x}_j \in \mathcal{S}_k} (x_{ji} - \mu_{ki})^2, \tag{3}$$

or

$$V = \sum_{k=1}^{K} \sum_{i=1}^{T} \sum_{\mathbf{x}_j \in \mathcal{S}_k} \left(x_{ji}^2 + \mu_{ki}^2 - 2 x_{ji} \mu_{ki}\right), \tag{4}$$

or

$$V = \sum_{k=1}^{K} \sum_{i=1}^{T} \left[\left(\sum_{\mathbf{x}_j \in \mathcal{S}_k} x_{ji}^2 \right) - |\mathcal{S}_k| \, \mu_{ki}^2 \right], \tag{5}$$

or finally

$$V = \sum_{k=1}^{K} \sum_{i=1}^{T} \left[\left(\sum_{\mathbf{x}_j \in \mathcal{S}_k} x_{ji}^2 \right) - \frac{1}{|\mathcal{S}_k|} \left(\sum_{\mathbf{x}_j \in \mathcal{S}_k} x_{ji} \right)^2 \right], \tag{6}$$

where we use the fact that $\mu_{ki} = \frac{1}{|\mathcal{S}_k|} \sum_{\mathbf{x}_j \in \mathcal{S}_k} x_{ji}$.

From the above derivation we observe two things:

- The objective function depends on the first ($\sum x_{ji}$) and second moment ($\sum x_{ji}^2$) of the data samples
- The result of the objective function depends on the object to cluster assignment ($\sum_{k=1}^{K} \sum_{i=1}^{T} (\cdot)$)

In the upcoming sections we will explicate a quantization scheme that (under certain assumptions) closely obeys the above two points.

3 Problem Specification and Approach

Our goal is to design a quantization scheme that retains the clustering structure as required by the K-Means algorithm. An illustration of this is shown in Figure 1. Consider the original data that consists of six time-series belonging to three different clusters. These clusters may be identified using the K-Means algorithm on the original data set. We

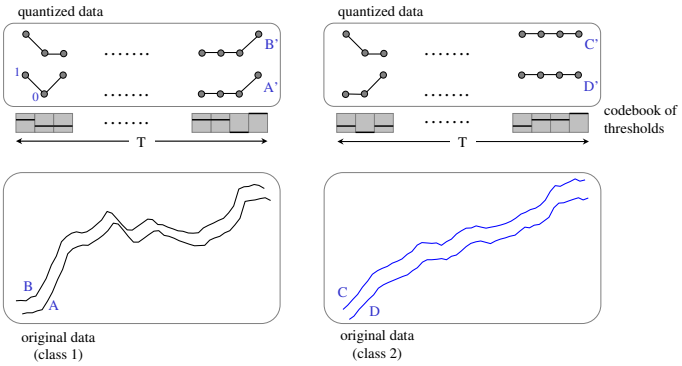

Figure 2. Overview of the proposed quantizer design

wish to design a quantization scheme such that the clustering structure is retained after quantization. Equivalently, if the K-Means algorithm is used to cluster the quantized time series, it should result in exactly the same clustering as obtained on the original time series.

In order to achieve the defined objective we take the approach shown in Figure 2. As shown, we design T scalar 1-bit quantizers per cluster, with one per dimension (or time instance, when dealing with time-series), i.e., one quantizer for the set of N_k 1-D samples on the same dimension (or co-located in time) for cluster k. Each quantizer is described in terms of a codebook, consisting of a single threshold, as well as two corresponding reconstruction levels. After quantization, the resulting signals may be viewed as binary 0-1 sequences. In the rest of the paper we show that we can build such 1-bit (single threshold) moment preserving quantizers that ensure that the simplified (quantized) dataset will result in identical clusters as the original (un-quantized) dataset, when using the K-Means algorithm.

4 Moment Preserving Quantization

Here we briefly review Moment Preserving Quantization (MPQ), which typically being used in the image processing literature [5] for retaining the texture properties of an image while providing good compression. We will present a quantizer that preserves the first two moments (i.e, mean and variance) for a set of samples, without making any assumptions on the distribution of the samples. We then derive two important observations for the proposed quantization:

- The quantization will preserve the mean and variance for any subset of samples.
- The quantization will lead to 'shrinking' of the original clusters under certain assumptions (which will be discussed later on)

We utilize these key observations in showing that the (optimal) K-Means clustering on the original and the quan-

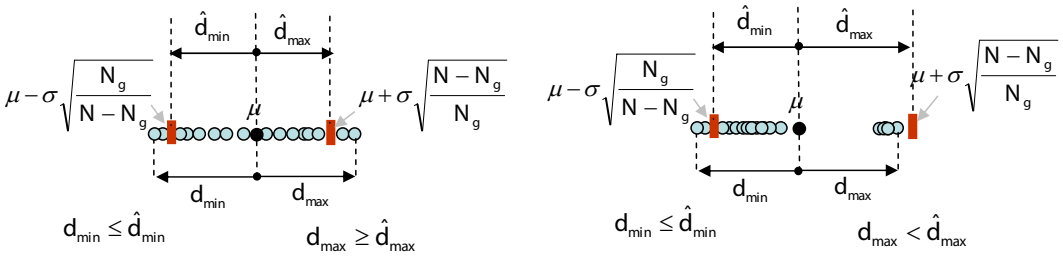

Figure 3. Different types of 1-D clusters

tized data will not change.

MPQ: The key idea behind MPQ may be described as follows. Consider a set of N 1-D samples x_j ($1 \le j \le N$) with sample mean μ and sample variance σ^2. The samples are transformed using the following 1-bit quantizer:

$$\hat{x}_j = \begin{cases} \mu + \sigma\sqrt{\frac{N-N_g}{N_g}} & x_j \ge \mu \\ \mu - \sigma\sqrt{\frac{N_g}{N-N_g}} & x_j < \mu \end{cases} \quad (7)$$

where N_g is the number of samples that have magnitude greater than or equal to μ. The resulting quantizer guarantees the preservation of the first two moments [5] of this set of samples i.e.

$$\frac{1}{N}\sum_{j=1}^{N}(x_j)^p = \frac{1}{N}\sum_{j=1}^{N}(\hat{x}_j)^p \text{ for } p = 1, 2. \quad (8)$$

We underscore that this is a *1-bit quantizer* since each sample can be replaced by a 0 or 1 indicating that it is above or below the mean value, since the mean value is preserved and hence can be reconstructed from the quantized values themselves.

More importantly for our discussion, we can show that this quantization leads to clusters that are tighter than before quantization. Formally, let the extent of the cluster be defined by $d_{max} = max_j\,(x_j - \mu)$ and $d_{min} = min_j\,(x_j - \mu)$. Similarly we can define $\hat{d}_{min} = min_j\,(\hat{x}_j - \mu)$ and $\hat{d}_{max} = max_j\,(\hat{x}_j - \mu)$. We first prove that we cannot have the cluster extent increase in both directions simultaneously, i.e., we cannot have $\hat{d}_{max} > d_{max}$ and $\hat{d}_{min} < d_{min}$ simultaneously. This is easy to show via contradiction. The moment preservation of the quantization implies

$$\sum_{j=1}^{N}(x_j - \mu)^2 = \sum_{x_j \ge \mu}\left(\hat{d}_{max}\right)^2 + \sum_{x_j < \mu}\left(\hat{d}_{min}\right)^2. \quad (9)$$

We use the fact that all points above the mean are quantized to the same value with distance $\hat{d}_{max}$ from the mean, and all points below the mean are quantized to the same value with distance $\hat{d}_{min}$ from the mean. If $\hat{d}_{max} > d_{max}$ and $\hat{d}_{min} < d_{min}$ we have,

$$\sum_{j=1}^{N}(x_j - \mu)^2 > \sum_{x_j \ge \mu}(d_{max})^2 + \sum_{x_j < \mu}(d_{min})^2 > N\sigma^2, \quad (10)$$

since $d_{max} \ge \sigma$ and $d_{min} \le -\sigma$ for any arbitrary data distribution. This is clearly a contradiction, since the LHS is equal to $N\sigma^2$.

Additionally, we show that the extent of the cluster in each direction does not increase due to quantization, except under some special conditions. If label the direction in which samples are greater than the mean as the *right* direction, and the direction in which samples are less than the mean as the *left* direction. Consider first the extent of the cluster in the right direction. For the cluster extent to increase after quantization, we need to have $\hat{d}_{max} = \sigma\sqrt{\frac{N-N_g}{N_g}} > d_{max}$, where we use the definition of the quantizer. The above means that the extent in the right direction can increase only when $N_g < N - N_g$ (this is a necessary, but not sufficient condition). Furthermore, as N_g decreases for a fixed N, it is more likely that quantization will lead to a greater increase in the right extent. The scenario when this does actually happen is when a cluster contains a small number of points that are far from the mean in the right direction, combined with a large number of points to the left of the mean. Intuitively, such a cluster is "ill-behaved" in that the data within it, actually belong to multiple sub-clusters that minimally overlap with each other. Ideally, this indicates that we can get better clustering performance by partitioning this into multiple sub-clusters. This idea is illustrated in Figure 3, where we show two 1-D cluster examples. On the left side is a "well-behaved" cluster that has a reasonably uniform spread, whereas on the right side is an ill-behaved cluster, where data to the right of the mean consists of very few but distant samples. In the figure we show the cluster samples, the mean, as well as the reconstruction levels in both directions. As shown, for the ill-behaved clusters it is possible that the extent of the

cluster increases in one direction after quantization.

In general, for well-behaved data clusters, we have $\hat{d}_{max} \leq d_{max}$. Similarly, for the left extent, we have $\hat{d}_{min} \geq d_{min}$ for well-behaved clusters.

5 K-Means and Moment Preserving Quantization

We now show how the MPQ scheme may be used to quantize data samples while retaining the clustering objective function of the K-Means algorithm. Given N time-series of length T each and the clustering result, i.e., subsets $\mathcal{S}_k(1 \ldots K)$, as determined by the outcome of the K-Means algorithm on the unquantized data, we build T 1-bit scalar quantizers per class. The scalar quantizer for cluster $\mathcal{S}_k$ operates on N_k samples (one per object) of the same dimension (or time for time-series data), and appropriately maps them into two bins. Equivalently we design one 1-bit quantizer per dimension of the N_k vectors clustered into cluster k. A value '0' represents that the time sample is below the quantizer threshold[1], while a value '1' represents that the sample is above the quantizer threshold. As a result of this quantization each time-series is then converted into a binary sequence of T 1's and 0's. Note that a 0 (or 1) may actually correspond to different reconstruction levels at different time instances.

5.1 Preservation of the K-Means Clustering Outcome

Recall equation 6 which is the expanded derivation of the K-Means objective function. For the new quantized values $\hat{x}_{ji}$ and given the properties of the moment preserving quantization, it is guaranteed that

$$\hat{V} = \sum_{k=1}^{K} \sum_{i=1}^{T} \left[\left(\sum_{\mathbf{x}_j \in \mathcal{S}_k} \hat{x}_{ji}^2 \right) - \frac{1}{|\mathcal{S}_k|} \left(\sum_{\mathbf{x}_j \in \mathcal{S}_k} \hat{x}_{ji} \right)^2 \right] = V. \tag{11}$$

What this means is that, given the cluster labels associated with each point, we can use 1-bit scalar moment preserving quantization across each dimension (time sample) of the vector (time series) within each cluster to guarantee that the resulting clustering metric is preserved.

Additionally, consider that we know the optimal set of clusters $\mathcal{S}_k^{opt}$, i.e., true clustering structure independent of the gradient descent Lloyd Algorithm for K-Means. This means that, for any other partitioning of the data into clusters $\mathcal{S}_k^*$, we have:

$$\begin{aligned} V^{opt} &= \textstyle\sum_{k=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{S}_k^{opt}} \sum_{i=1}^{T} (x_{ji} - \mu_{ki})^2 \\ &\leq \textstyle\sum_{k=1}^{K} \sum_{\mathbf{x}_j \in \mathcal{S}_k^{*}} \sum_{i=1}^{T} (x_{ji} - \mu_{ki})^2 = V^* \end{aligned} \tag{12}$$

[1]For *Moment-Preserving-Quantization* this is the mean of the N_k samples.

If we now use the scalar moment preserving quantization, designed based on the optimal clustering labels $\mathcal{S}_k$, we can show that.

$$\hat{V}^{opt} = V^{opt} \leq V^* \text{ and } \hat{V}^{opt} = V^{opt} \leq \hat{V}^* \tag{13}$$

The moment preserving property of the quantization ensures that $\hat{V}^{opt} = V^{opt}$ while $V^{opt} \leq V^*$ by definition of the optimal clustering scheme. Additionally, we know that for "well-behaved" clusters, quantization actually results in shrinking the cluster extent towards the mean, making them tighter. As the mean squared error (MSE) optimal K-Means clustering uses nearest-neighbor assignment of data samples to cluster centroids, the tighter clusters are guaranteed to retain the optimality of the original clustering scheme.

This is an important result as it implies that if we quantize the original time series using a set of 1-bit moment preserving quantizers per cluster, designed based on the optimal clustering labels, the new set of quantized samples retains the same optimal clustering structure. Alternatively, if we take this new set of quantized time series, and cluster them using the K-Means algorithm, they will result in the same cluster labels as for the original unquantized data set[2].

5.2 Compression Efficiency

Here we analyze the compression efficiency of the proposed quantization scheme. For the set of N object with T dimensions clustered into K clusters, let each unquantized sample be represented by B bits. Then the total storage requirement is BTN bits for the unquantized data, and $N \log_2(K)$ bits to indicate the clustering labels - with $\log_2(K)$ bits per object.

Instead, if we use a 1-bit quantizer we need to store only TN bits for all objects, along with $2BTK$ bits to store the two reconstruction levels per dimension per cluster. Note, that the threshold does not need to be explicitly stored as it can be deduced from the reconstructed samples, since the quantization does not distort the mean. Finally, $N \log_2(K)$ bits are also required to indicate the clustering labels. Hence the compression efficiency ρ achieved by our moment preserving quantization scheme is:

$$\rho = \frac{\text{bytes quantized}}{\text{bytes unquantized}} = \frac{TN + 2BTK + N \log_2(K)}{BTN + N \log_2(K)}. \tag{14}$$

Since typically we have $N - 1 > 2K$ the compression efficiency $\rho < 1$.

A better compression ratio can be achieved by noting that since the quantization preserves the underlying clustering structure, one does not explicitly need to store the cluster

[2]This discussion ignores the gradient descent nature of the algorithm that sometimes gets trapped in local minima.

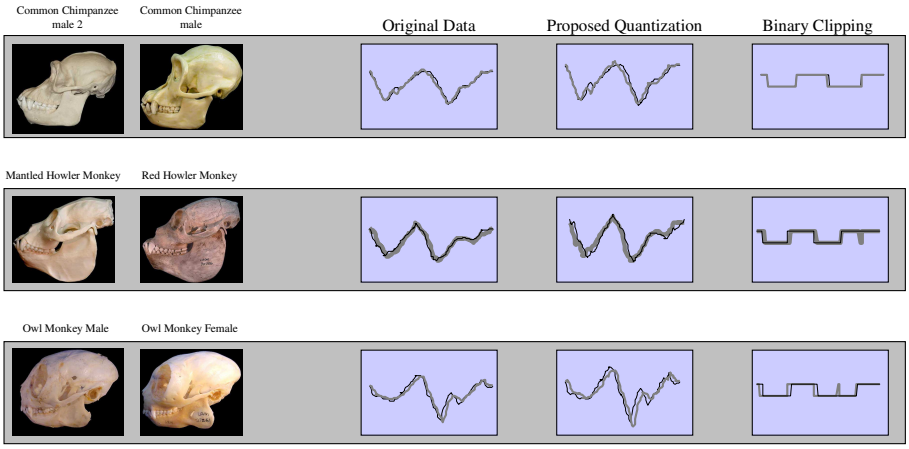

Figure 4. Shape preservation of quantization techniques. From left to right: Skull shapes, original extracted sequences, sequences after proposed quantization, sequences after the 'clipping' approach of [3]

labels. Then, while we require BTN bits for the original data set, for the quantized data sets only $\log_2(2TK)TN$ bits are sufficient. This is because there are only 2 possible values that each of the T samples can take per cluster, and K clusters, i.e., a total of at most $2TK$ values. Hence we need at most $\log_2(2TK)$ bits per sample. The corresponding no label compression efficiency $\hat{\rho}$ may be defined as:

$$\hat{\rho} = \frac{\log_2(2TK)}{B}. \tag{15}$$

In the experimental section 6.4 we evaluate on real datasets the compression efficiency of the quantization scheme.

We also note that, while other quantization schemes such as the the clipping approach of [3] require just B bits to represent the temporal mean of each of the N object, and then $2T$ bits per time sample, i.e., a total of $N(B+T)$ bits, this however comes at the expense of post quantization clustering accuracy. In the experiments show that our technique achieves superior clustering performance on the quantized data compared to such approaches.

5.3 Discussion

While it is easy to see that a trivial cluster-preserving compression can be achieved by retaining only cluster centroids and the label for each vector and distributing just those, it is nonetheless apparent that this compression is very lossy and destroys distinctions among objects within the same cluster. Therefore such an approach would have limited the use of the compressed dataset in other operations such as visualization or Nearest-Neighbor search. Instead, in our approach we can control the granularity of compression, and show that even with 1-bit quantization the compressed data samples retain "shape" as well as neighborhood relationships. We depict that neighborhood is objects is well preserved in the experimental section. Finally, we note that pre-clustering cluster preservation techniques, such as [10, 11], can achieve cluster preservation and data obfuscation, but they are not designed for data compression or shape preservation. Such approaches completely change the 'shape' of the original data by transforming them into a different domain.

Our K-Means cluster preservation technique is designed for Euclidean based separable distance functions between objects. However, in many high-dimensional datasets (e.g., for time-series) warping distance functions and other non-separable functions are also typically used. While our technique does not carry over to such distance measures, in Section 6.3 we show sample results of our technique on *phase (rotation) invariant* distance measures, the outcome of which closely resembles that of warped distance functions. Extensions to support other types of distance measures are under consideration.

6 Experiments

In this section we validate the performance of our quantization on real data sets. We examine various characteristics of the quantization scheme including its effect on the 'shape' of the data, quality of cluster preservation and we also delineate preliminary results for for neighborhood (k-NN) classification. We utilize stock market time-series data, as well as 2-dimensional shape contour data. The 2D contours come from three datasets, representing `skulls`, `fish` and `leaves` shapes. We convert the 2D shapes into 1D sequences by finding the center of mass and extracting the distance to all perimeter points. Such 1D sequence features are commonly utilized in shape indexing and search experiments [7].

6.1 Shape Preservation

Here we demonstrate that the proposed 1-bit quantization does in fact retain the with high accuracy the object 'shape'. We present results using the 1D sequences extracted from the `skulls` dataset. Figure 4 depicts 6 skulls with extracted 1D sequences, the quantized representation and the 'clipped' representation as presented in the work of [3]. In our quantized representation, we use the reconstruction levels of each quantizer instead of the '0' or '1' values. We observe that the proposed moment-preserving quantization retains very closely the form of the original sequences. On the other hand the clipped representation captures the data fluctuations, but due to the purely binary representation cannot discriminate the difference between the relative amplitudes at different positions of the sequence.

6.2 Cluster preservation

Now we examine how well the clusters are retained on the quantized dataset. We utilize time series from stock market data corresponding to 1800 stock symbols from companies listed on Nasdaq, reporting the stock values for a period of approximately 3 years.

Our theoretical results about cluster preservation are applicable for the optimal K-Means algorithm. Since the exact solution is NP-hard to compute, in practice a gradient descent (Lloyd's algorithm) variation is used for computational simplicity. Therefore, here we empirically evaluate the discrepancy of the clustering results, when the non-exact algorithm is utilized.

Since, we do not have the cluster labels for aforementioned dataset, we cluster these time-series into 8 clusters using the Lloyd Algorithm for K-Means[3]. As we desire a near-optimal set of clusters, we repeat the K-Means algorithm with multiple starting points and select the set that achieves the smallest K-Means metric. Note that we can also use algorithms such as K-Means++ [1] to achieve a very good initial estimate of the cluster centroids and maximize the probability of finding the global optimum. We then use this clustering structure as ground truth, and quantize the time series from each cluster using a separate moment-preserving 1-bit quantizer. The resulting cluster centroids, sample time series from each cluster, and the upper and lower reconstruction levels for each 1-bit quantizer are shown in Figure 5.

Cluster centroids are shown in black, while 3 example series from each cluster are shown in gray. The 1-bit quantizer levels for each sample for each cluster are shown in red. As is clear, the quantizer does tend to follow the "mean" shape of the cluster quite nicely. We then rerun the clustering using Lloyd's algorithm on the quantized time series, using the same initial conditions (the same sample points before and after quantization were chosen as the initial cluster centroids) and compare the resulting trained cluster centroids before and after quantization. These centroids are shown in Figure 6. As may be seen from the figure, a majority of the cluster centroids are almost identical before and after quantization. This is however not true for clusters 3 and 6. There are two reasons for this, clusters 3 and 6 are the least well-behaved of the clusters, and both violate the $\hat{d}_{max} > d_{max}$ condition. Additionally, as the Lloyd algorithm is gradient descent based, while the optimal clustering structure remains before and after quantization, it is possible that an intermediate sub-optimal clustering, as demonstrated here, may be different. However, even for this data set, quantization results in the mis-labeling of only 57 of the 1800 time-series ($\sim 3\%$), a majority of which (49) belong to these two clusters. The resulting confusion matrix, comparing unquantized and quantized labels is:

$$\begin{bmatrix} 378 & 1 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 169 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 89 & 0 & 0 & \mathbf{43} & 0 & 0 \\ 0 & 0 & 0 & 98 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 329 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{6} & 0 & 1 & 129 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 353 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 198 \end{bmatrix},$$

where the column index corresponds to the label before quantization and the row index represents the label after quantization. The confusion is primarily restricted to data from clusters 3 and 6, and this is also reflected in Figure 6.

For the same dataset we use the clipped data representation [3] to quantize the time series into 1 bit per sample and then redo the clustering. Clipping is performed per time series and replaces a sample with '1' if it lies above the mean (computed across the time samples of that series) and '0' if it lies below. Clipping thus does not require any prior knowledge of the data clusters. However, this results in a significant impact on the label preservation performance. For the stock data set, the clustering after clipping results in only 203 time-series retaining their label - *less than 20% of the data*. Hence, clipping not only retain as well the object shape, but also performs significantly worse the cluster operations of the quantized data.

Sensitivity to initial centers: Here we evaluate the sensitive to the selection of the right starting point (the seed centroids). In order to quantify the impact of mismatched starting points on clustering before and after quantization, we use a set of disjoint starting points for the two cases. The resulting mismatch is quantified in terms of the number of signals mislabeled. We find that for a set of 10 different starting points, the mean mismatch was 140 (7.8%) signals, with a maximum of 230 (12.7%) signals. Additionally, in

[3]The number of clusters was selected manually based on the achieved quantization error, and visual inspection of the time series

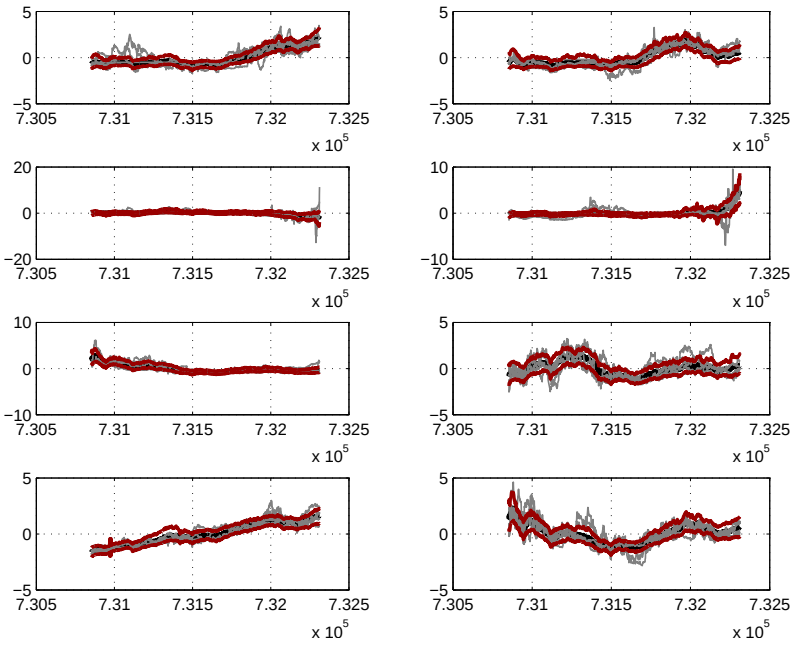

Figure 5. Sample time series from stock data grouped into 8 clusters. Clusters numbered left to right and top to bottom.

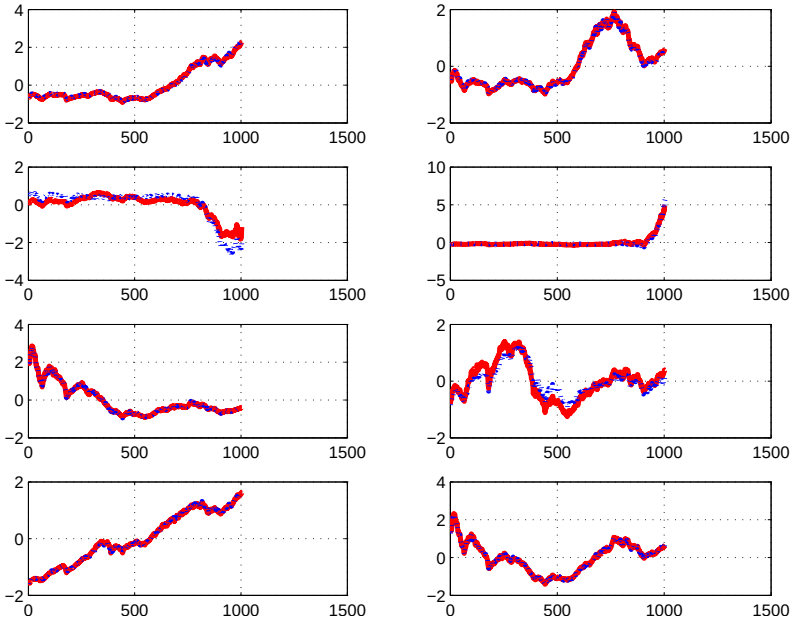

Figure 6. Cluster centroids before (red) and after (blue) 1-bit quantization. Clusters numbered left to right and top to bottom.

most cases cluster centroids are very close before and after quantization. This low level of mismatch indicates that the clustering structure is well retained after quantization.

Neighborhood preservation: While the quantization scheme is not designed for k-nearest neighbor (k-NN) classification, the fact that it tends to retain the clustering structure indicates that it may also preserve results for k-NN - if the class labels are assumed to be determined by the clustering structure. In order to verify this, we assume that the unquantized clusters represent ground truth with 8 labels for the data. We use k-NN with varying numbers of neighbors and use majority voting among neighbors to determine the predicted class label. We present classification results on the training set before and after quantization in Table 1.

The limited separation between the clusters ensures that increasing the number of neighbors does not always result

Table 1. k-NN classification accuracy before and after quantization

	5-NN	10-NN	20-NN
Unquantized	97.3%	96.9%	95.4%
Quantized	91.1%	91.2%	90.4%

in increasing the prediction accuracy[4]. However, importantly, 1-bit quantization degrades the k-NN performance only by 5–6% in this scenario. While the scheme does not provide any guarantees on nearest neighbor classification, this is a useful illustrative example to show the performance on real data. Attempting to bound performance of this scheme for the k-NN classification scheme is a direction for future work.

6.3 Contour Data Sets

In this section we repeat the clustering experiments for the `fish` and the `leaves` data sets, both of which represent the 2D contours of the corresponding images. The `fish` data set consists of 247 contours, while the `leaves` data set contains 1125 contours. These contours are converted into series of samples by extracting the distance of the perimeter points from the center of mass. Additionally, when one wishes to support rotation invariance, their periodogram can be extracted and used as the sequence feature, similar to the method used in [14, 13]. The new rotation-invariant sequences can now be used to provide more flexible clustering results. Utilizing the `fish` dataset we demonstrate some of those clustering results in Figure 7. Observe that rotated versions of the same shape are now clustered together. Obviously, various erroneous object placements can be detected in the figure as well, however, this example serves as another demonstration of the meaningfulness of time-series clustering using K-Means.

As we do not have ground truth labels for the `fish` and `leaves` datasets, we perform clustering with different number of clusters for both data sets. We present the resulting label preservation in terms of the number of series that retain their label after quantization. These results are shown in Table 2. In the table, we also present the results for clipping.

Table 2. Label Preservation with Quantization

Dataset	Scheme	$K = 5$	$K = 10$	$K = 15$
`fish`	MPQ	96.3%	98.4%	85%
	Clipping	74.4%	75.7%	70.4%
`leaves`	MPQ	79.1%	88.6%	96.8%
	Clipping	25.6%	33.2%	34.2%

From the table, we see that the labels for the `fish` data set are very well preserved by the proposed moment preserving quantization (MPQ), with labels retained for 98.4% of the series when the number of clusters is 10. Note that reducing the number of clusters to 5 leads to the creation of more "ill-behaved" clusters, i.e., data from multiple clusters gets forced into one umbrella cluster, and hence the label preservation is not as good. Additionally, as the number of clusters increases beyond a certain level, there is insufficient data to train the clusters, and hence identifying 15 clusters from the 247 `fish` series leads to poor clustering, and as a result poor label preservation. As opposed to the `fish` data set, the `leaves` data set has continuously increasing performance with the number of clusters. This is because the data set contains a much larger number of series, avoiding the problem of overfitting. Furthermore, as the number of clusters decrease, there is a higher probability of "ill-behaved" clusters reducing performance. The best results are achieved for this data set are when $K = 15$ when 96.8% of the series retain their label. Also, from the table we see that MPQ always outperforms Clipping, which performs reasonably on the simpler `fish` data set - 76% of the series retain the same label before and after quantization - but performs significantly worse on the `leaves` data set - only 34% of the samples retain their label. Finally, in terms of sensitivity to the mismatch in initial centroid selection for quantized and unquantized datasets, the standard deviation for MPQ on the `fish` data set is 7.2%, while it is 6.8% for the `leaves` dataset.

6.4 Compression Efficiency

Lastly, we evaluate the compression efficiency of the quantization scheme. We report the compression efficiency $\hat{\rho}$, as defined in equation (15), for the datasets used in our experiments. $\hat{\rho}$ is reported as percentage of the quantized dataset compared to the original one.

Table 3. Compression Efficiency

Dataset	N	Dim	K	$\hat{\rho}$
`stock`	1800	1004	8	44%
`fish`	247	256	5	35%
			10	38%
			15	40%
`leaves`	1125	128	5	32%
			10	35%
			15	37%

For computing these values, we assume that the unquantized data samples are represented by 4 bytes each, i.e., $B = 32$. As can be seen from the table, the compression can be as small as 35%, a reduction in size by almost a factor of 3. The compression efficiency varies with the number of clusters, deteriorating as K increases. Selecting the optimal trade-off between compression, clustering, and cluster label preservation is an important practical consideration for this scheme.

[4]This may also be because the labels are not true class labels, but are derived from clustering.

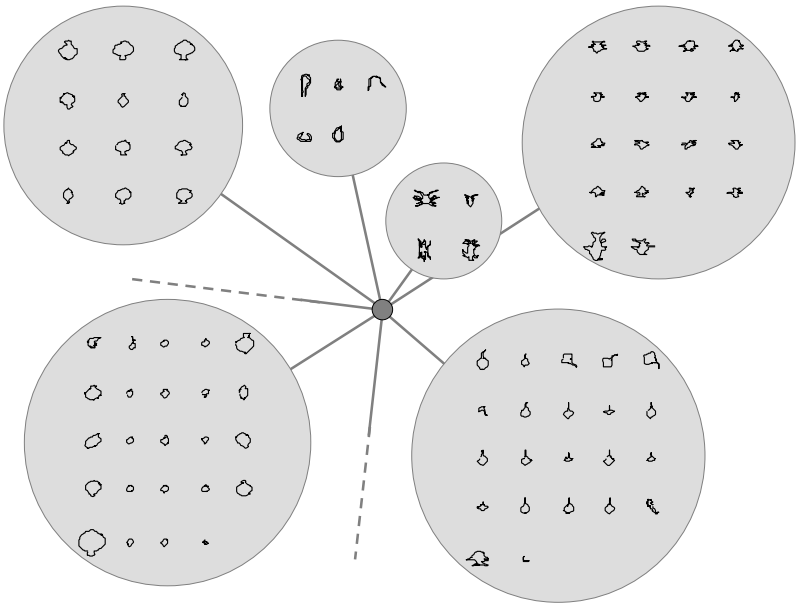

Figure 7. Some of the discovered clusters for the `fish` dataset in conjunction with rotation invariant time-series features

7 Conclusion

We showcased compression schemes for high-dimensional data sets that preserve the outcome of K-Means clustering. Our analytic derivation indicates that a quantizer that retains the first two moments of each dimension of the data set, per cluster, does not change the optimization metric for K-Means and therefore can guarantee preservation of the underlying cluster structure. Such a quantizer can be designed using 1-bit per dimension Moment Preserving Quantization. As future work, we plan to investigate the design of multi-bit quantizers, for providing fine-grained trade-offs between clustering guarantees and shape preservation. We also propose to investigate the interaction with transform domain (Wavelet, Fourier) based compression schemes and design extensions for non-separable distance functions.

References

[1] D. Arthur and S. Vassilvitskii. k-Means++: The Advantages of Careful Seeding. In *Proc. of Symposium of Discrete Analysis*, 2005.

[2] F. Bach and M. Jordan. Learning spectral clustering. In *Proc. of NIPS*, 2004.

[3] A. J. Bagnall, C. A. Ratanamahatana, E. J. Keogh, S. Lonardi, and G. J. Janacek. A Bit Level Representation for Time Series Data Mining with Shape Based Similarity. In *Data Min. Knowl. Discov. 13(1)*, pages 11–40, 2006.

[4] P. S. Bradley, U. M. Fayyad, and C. Reina. Scaling Clustering Algorithms to Large Databases. In *Proc. of SIGKDD*, pages 9–15, 1998.

[5] E. Delp, M. Saenz, and P. Salama. Block Truncation Coding (BTC). In *The Handbook of Image and Video Processing*. Academic Press, 2000.

[6] G. Jagannathan and R. N. Wright. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proc. of SIGKDD*, pages 593–599, 2005.

[7] E. J. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos. Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures. In *Proc. of VLDB*, pages 882–893, 2006.

[8] J. Lin, M. Vlachos, E. J. Keogh, and D. Gunopulos. Iterative Incremental Clustering of Time Series. In *Proc. of EDBT*, pages 106–122, 2004.

[9] V. Megalooikonomou, G. Li, and Q. Wang. A dimensionality reduction technique for efficient similarity analysis of time series databases. In *Proc. of CIKM*, pages 160–161, 2004.

[10] S. R. M. Oliveira and O. R. Zaane. Privacy Preservation When Sharing Data For Clustering. In *Intl. Workshop on Secure Data Management in a Connected World*, 2004.

[11] R. Parameswaran and D. Blough. A Robust Data Obfuscation Approach for Privacy Preservation of Clustered Data. In *Workshop on Privacy and Security Aspects of Data Mining*, pages 18–25, 2005.

[12] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proc. of SIGKDD*, pages 206–215, 2003.

[13] M. Vlachos, Z. Vagena, P. S. Yu, and V. Athitsos. Rotation invariant indexing of shapes and line drawings. In *Proc. of CIKM*, pages 131–138, 2005.

[14] M. Vlachos, P. S. Yu, and V. Castelli. On Periodicity Detection and Structural Periodic Similarity. In *Proc. of SDM*, 2005.